

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 60-117335

(43)Date of publication of application : 24.06.1985

(51)Int.Cl.

G06F 9/38

G06F 9/28

(21)Application number : 58-224130

(71)Applicant : HITACHI LTD

(22)Date of filing : 30.11.1983

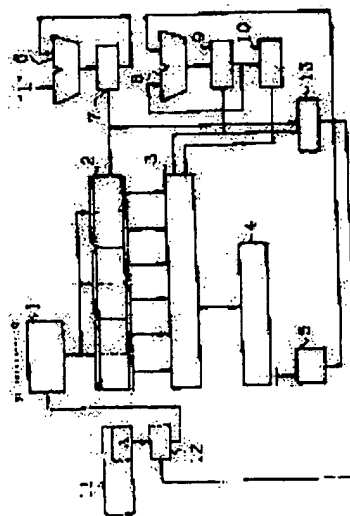
(72)Inventor : YU KEIICHI  
TSUNEHIRO TAKASHI  
NAKAKOSHI JUNJI  
NAKAI KOICHI

## (54) INFORMATION PROCESSOR

### (57)Abstract:

**PURPOSE:** To obtain an advance control type information processor which can reduce the quantity of hardware without deteriorating performance, by the full data length of an instruction buffer register at the value shorter than (m) times as much as the maximum instruction word and equal to (n) times as much as the instruction reading length.

**CONSTITUTION:** An instruction buffer register 2 stores the present and following instructions read out of a main memory 1 at a position shown by an instruction storing pointer 7. A selector 3 selects and extracts the instructions within the register 2 which are designated by a present instruction position pointer 10 and a following instruction position pointer 9 and sends them to an instruction analysis register 4. An instruction word length decoder 5 produces the following instruction word length and sends it to an adder 8. The pointer 9 adds the following instruction word length produced from the decoder 5 by an operator 8 when a microinstruction executes the next instruction. The result of this addition is delivered to the pointer 9 after the contents of the pointer 9 are shifted to the pointer 10 and at the same time supplied to the adder 8.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

Japanese Publication No. 60-117335  
(translation)

Patent Publication (Laid-Open) No. 60-117335

Publication Date: June 24, 1985

Patent Application No. 58-224130

Filing Date: November 30, 1983

Inventor(s): TSUNEHIRO Takashi, NAKAGOSHI Junji

Applicant(s): HITACHI

### Specification

#### 1. Title of the Invention

Information processing unit

#### 2. Claims for the Patent

(1) An information processing unit comprising a plurality of instruction buffer registers for precedent-controlling, a pointer indicating a location for storing the next instruction in the register, and a pointer indicating a location for reading from the register, characterized by comprising means for determining whether or not the next instruction can be read in the instruction buffer register by comparing each of pointers indicating locations for reading a current instruction and a successive instruction from said instruction buffer register and a value of a pointer indicating the location for reading a successive instruction and a value of a pointer indicating said location for storing the next instruction, and controlling means for determining whether or not reading of the next instruction should be canceled in accordance with said determination means and indicating to

Japanese Publication No. 60-117335  
(translation)

unconditionally read the instruction if the next instruction is an decision instruction.

### 3. Detailed Description of the Invention

#### [Field of the Invention]

The present invention relates to an information processing unit, and more particularly to a precedent-controlling information unit with a micro program that can reduce an amount of hardware without lowering performance.

#### [Background Art]

Precedence controlling is performed for a higher speed operation of an information processing unit. In this case, a data length for an instruction word reading register for precedently reading an instruction word needs to be at least  $(\text{maximum instruction word length}) \times (\text{the number of words to be precedently read} + 1)$ .

In other words, an instruction buffer register of a conventional precedent-controlling information processing unit includes, for example, unit instruction registers IR1 - IR4 as shown in Figure 1. In Figure 1, L1 is the minimum instruction word length, L2 is the data length read from memory and L3 is the maximum instruction word length.

As shown, an instruction buffer register includes a plurality of registers IR with a data length of L2 (data length read from memory), which is n times (here n is a positive integer) the minimum instruction word length L1, and the total data length of an instruction buffer register is m times (here m is a positive integer) the maximum instruction

Japanese Publication No. 60-117335  
(translation)

word length  $L3$  (in the case of Figure 1,  $n = m - 2$ ). This enables precedent controlling even when instructions of the maximum instruction word length  $L3$  comes in a row, because the next instruction is stored in the instruction buffer register.

However, there is a problem in an information processing unit that has low frequency of usage of an instruction of the maximum instruction word length  $L3$ , even if the total data length of an instruction buffer register is multiple times the maximum instruction word length  $L3$ , the information processing unit has no effect of speeding up the processing of an instruction with a shorter length than the maximum instruction word length  $L3$  and higher frequency of usage, therefore, the information processing unit has not fully made use of the instruction buffer register.

[Problems to be solved by the Invention]

The present invention intends to provide an information processing unit that is downsized and price-reduced without lowering its performance even for an instruction of lower frequency of usage and longer instruction word length by reducing the amount of hardware to solve the above mentioned conventional problem.

[Means for Solving the Problems]

In order to achieve the abovementioned object, an information processing unit according to the present invention is an information processing unit including a composite instruction buffer register for precedent-controlling, a pointer indicating a location for storing the next instruction

Japanese Publication No. 60-117335  
(translation)

in the register, and a pointer indicating a location for reading from the register, characterized by including means for determining whether or not the next instruction can be read in the instruction buffer register by comparing each of pointers indicating locations for reading a current instruction and a successive instruction from said instruction buffer register and a value of a pointer indicating the location for reading a successive instruction and a value of a pointer indicating said location for storing the next instruction, and controlling means for determining whether or not reading of the next instruction should be canceled in accordance with said determination means and indicating to unconditionally read the instruction if the next instruction is a decision instruction.

[Embodiment of the Invention]

An embodiment of the invention will be described with reference to drawings.

Figure 2 is a block diagram of an instruction buffer register used for the present invention, and Figure 3 is a block diagram of an information processing unit showing an embodiment of the present invention.

As shown in Figure 2, in the embodiment, an instruction buffer register with two bytes for the minimum instruction word length L1, four bytes for the reading instruction word length L2, eight bytes for the maximum instruction word length, and 12 bytes for the total data length of the instruction buffer register. In this case, the amount of hardware is

Japanese Publication No. 60-117335  
(translation)

reduced by  $1/4$  of that of the conventional instruction buffer register.

Figure 3 shows an instruction buffer register 2 shown in Figure 2. In Figure 3, reference number 1 denotes main memory, 3 denotes an instruction selector, 4 denotes instruction analysis register, 5 denotes instruction word length decoder, 6 and 8 both denote calculators, 7 denotes a storage location pointer for the next instruction, 9 denotes a successive instruction reading location pointer indicating the head, 10 denotes a current instruction reading location pointer indicating the head, 11 denotes a micro instruction register, 12 denotes a main memory controlling circuit and 13 denotes an instruction reading restraint condition circuit.

First, the instruction buffer register 2 stores two instructions (a current instruction and a successive instruction) read from the main memory 1 in the location indicated by the instruction storage pointer 7. As the instruction storage pointer 7 is formed by two bits, it can indicate three instruction storage locations (in units of 12) in the instruction buffer register 2 with "00", "01" and "10" ("11" is not used here).

When an instruction is stored in the instruction buffer register 2, calculator 6 adds 1 so that the storage location is updated. In this case, the calculator 6 is an addition circuit, which is adapted to output "00" when a result of addition is "11".

Japanese Publication No. 60-117335  
(translation)

The instruction selector 3 selects each of the instructions in an instruction buffer register 2 indicated by the current instruction location pointer 10 and the successive instruction location pointer 9, retrieves each of the instruction words in order and store it in the instruction analysis register 4 at the next stage. When the instruction is stored in the instruction analysis register 4, the instruction word length decoder 5 decodes an instruction word length bit in the instruction analysis register 4, generates an instruction word length of the successive instruction and sends it to the adder 8.

When a micro instruction executes the next instruction word, the successive instruction location pointer 9, formed by three bits, makes the calculator 8 to add an instruction word length of the successive instruction generated by the instruction word length decoder 5 and outputs the result to the successive instruction location pointer 9. Prior to this, contents stored in the successive instruction location pointer 9 are transferred to the current instruction location pointer 10, while the contents are added to as another input to adder 8.

A successive instruction location pointer can indicate six instruction storage locations (in units of L1) in an instruction buffer register 2 with "000", "001", "010", "011", "100" and "101" (as "110" and "111" are not used here, calculator 8 outputs "000" when the result of addition is "110"). In this case, the two higher-order bits of the three

Japanese Publication No. 60-117335  
(translation)

bits indicate a location in units of L2 separated by a solid line, while the one lower-order bit indicates a before or after location in units of L1 separated by a dotted line.

The main memory 1 stores four bytes (32 bits) for each address, though an instruction unit can be the types of two bytes (16 bits), four bytes (32 bits) and eight bytes (64 bits). Therefore, when a successive instruction is precedently read from the main memory 1 along with a current instruction, all the two instructions are stored in the instruction buffer register 2 or only a part of the successive instruction is stored with the rest part still being stored in the main memory 1.

The embodiment can be implemented without significantly lowering the performance in every case that (I) it executes the successive instruction, while all the instruction precedently read being stored in the instruction buffer register 2 (in the case of a typical step instruction), (II) it cancels the precedently read instructions and performs another reading to the same location even if all the precedently read instructions are stored (when the next instruction is a decision instruction), and (III) it reads a successive instruction in a location of the current instruction and executes address calculation for the successive instruction, while a micro instruction is executing the current instruction, as all the instructions to be precedently read cannot be stored in the instruction buffer



Japanese Publication No. 60-117335  
(translation)

register 2 (when an amount of the buffer register is not enough).

In Figure 3, precedent-controlling is executed according to the micro instruction. In the case of a typical unconditional branch, an address part of the micro instruction register 11 is transferred to a controlling storage address register and a particular one bit of the micro instruction register 11 is set to "0" or "1" as a branch indication bit. In the case of a conditional branch, if the condition is met as a result of the branch test, an address part of the micro instruction register 11 is transferred to a controlling storage address register, and if the condition is not met, the contents of the controlling storage address register are incremented by 1 and the process continues to control the normal step.

When one bit of an instruction reading controlling part A of the micro instruction register 11 is "0", the memory controlling circuit 12 controls to unconditionally read an instruction from the main memory 1 (in the case of a branch instruction). An indication by this micro instruction will be called IF1 (Instruction Fetch 1). When one bit of an instruction reading controlling part A of the micro instruction register 11 is "1" and if an output from the instruction reading restraint condition circuit 13 is "0", the memory controlling circuit 12 performs reading of an instruction, and if it is "1", it controls to cancel the reading of an instruction. An indication by a micro

Japanese Publication No. 60-117335  
(translation)

instruction to cancel as mentioned above will be called IF2 (Instruction Fetch 2).

The instruction reading restraint condition circuit 13 compares two bits of the instruction storage pointer 7 and the two higher-order bits in three bits of the successive instruction location pointer 9. If they match, it outputs "1". Otherwise, it outputs "0". If an output from the instruction reading restraint condition circuit 13 is "1", it means that the location to store the next instruction reading matches a location where a successive instruction is stored. In other words, if a micro instruction indicates IF1, it means that the next instruction word is canceled to be stored in the location where the successive instruction is stored and an instruction additionally stored will be executed first (e.g., branch instruction or the like). If a micro instruction indicates IF2, it means that the next instruction word is canceled to be stored in the location where the successive instruction is stored and the abovementioned successive instruction that is already stored will be executed first.

Figure 4 is a flow chart of a micro program for controlling the operation shown in Figure 3.

First, in order to read the top instruction, the micro instruction sets an instruction address in address register (not shown) and indicates IF1 for three times (because of three instruction buffer registers in units of L2, steps 101, 102). At this moment, the instruction storage pointer 7 is "00" when an address register is set and stores instruction

Japanese Publication No. 60-117335  
(translation)

words from the top of the instruction register 2 in order (step 103). Also, the successive instruction pointer 9 sets "000" when the instruction address is even and sets "001" when it is odd (step 104). When the instruction is read and stored in the instruction buffer register 2, the micro instruction indicates an instruction code branch and branches to the top micro address of the micro instruction that performs the instruction (step 106). Before the instruction code branch, the instruction selector 3 reads an instruction from a location indicated by the successive instruction location pointer 9 and stores it in the instruction analysis register 4 (step 105). The instruction word length decoder 5 decodes the instruction word length and outputs the instruction word length to the calculator 8 that in turn adds the contents of the successive instruction location pointer 9 that is previously set and the instruction word length (step 107). When a micro instruction indicates an instruction code branch, the successive instruction location pointer 9 stores the result of addition (step 108). At the same time, the current instruction location pointer 10 stores the contents of the previous successive instruction location pointer 9 (step 109). A micro instruction that is branched by the instruction code indicates IF1 and IF2 as follows in accordance with the instruction word length and indicates to execute the current instruction (step 110).

(I) When an instruction word length is two bytes or four bytes, the micro instruction indicates IF2 once (step 112).

Japanese Publication No. 60-117335  
(translation)

This is because, in the case that the successive instruction that is already stored is of the maximum word length L3, if the instruction is shorter than four bytes, it can store the next instruction.

(II) When an instruction word length is six bytes or eight bytes, the micro instruction indicates each of IF1 and IF2 once (step 113). That is to say, when the stored successive instruction is equal to or shorter than four bytes length, it issues IF1 because the successive instruction can be read and stored in the remaining registers. When the stored successive instruction is equal to or longer than six bytes, it issues IF2 and makes the next reading to be cancelled because the successive instruction cannot be stored in the remaining registers.

(III) Without regard for an instruction word length, the micro instruction indicates IF1 for three times in the case of a decision instruction (step 114). In this case, it unconditionally reads and stores and executes the branched instruction. It indicates IF1 for three times because there are three instruction buffer registers in units of L2.

When a micro instruction executes the current instruction and it no longer refers to the contents of instruction analysis register 4, i.e., the next instruction word length is longer than the rest part of the register and the next instruction cannot be read, it indicates IF1 or IF2 to indicate operand address calculation of the successive instruction and reading of the operand (step 115). In this

Japanese Publication No. 60-117335  
(translation)

case, as the successive instruction indicates IF2, the instruction selector 3 stores the successive instruction in the instruction buffer register 2 from a location indicated by the successive instruction location pointer 9 into the instruction analysis register 4 in accordance with a precedence address calculation indication, performs an operand address calculation of the successive instruction and reads the operand according to the calculated address, without being broken by the next instruction reading. When the current instruction has two bytes length, it can calculate the precedent operand address of the successive instruction before the completion of the instruction reading. When the current instruction has four bytes length and the successive instruction has longer than six bytes length, it does not perform the precedent operand address calculation until the instruction reading finishes, as the instruction buffer register 2 stores only a part of the instruction (step 116).

When the current instruction has six bytes length or eight bytes length and the instruction buffer register 2 stores only a part of the successive instruction, it nether performs the precedent operand address calculation of the successive instruction until the instruction reading finishes, as in the above manner.

When only a part of the abovementioned successive instruction is stored, the micro instruction reduces the actual waiting time by executing the current instruction, while waiting for the completion of the instruction reading.

Japanese Publication No. 60-117335  
(translation)

In this manner, in the embodiment, the total data length of the instruction buffer register is shortened to reduce cost, and the instruction of short instruction word length is subject to a high speed operation, while an instruction is enabled to be executed without significantly lowering the performance even the instructions of longer instruction word length come in row.

[Advantages of the Invention]

As mentioned above, according to the present invention, the total data length of the instruction buffer register can be made shorter than  $m$  times (e.g., twice) the maximum instruction word and equal to the data length of  $n$  times (e.g., triple) the instruction reading length. The reduced amount of hardware enables downsizing and cost reduction. Moreover, the performance is not so much lowered in the present invention than in the conventional arts.

4. Brief Description of the Drawings

Figure 1 shows a block diagram of instruction buffer register in the conventional art.

Figure 2 shows a block diagram of instruction buffer register in the present invention.

Figure 3 shows a block diagram of a substantial part of an information processing unit illustrating an embodiment of the present invention.

Figure 4 shows a flow chart of an operation of a micro program for controlling Figure 3.

[Description of Symbols]

Japanese Publication No. 60-117335  
(translation)

- 1: main memory
- 2: instruction buffer register
- 3: instruction selector
- 4: instruction analysis register
- 5: instruction word length decoder
- 6, 8: calculator
- 7: instruction storage pointer
- 9: current instruction location pointer
- 10: successive instruction location pointer
- 11: micro instruction register
- 12: memory controlling circuit
- 13: instruction reading restraint condition circuit

Figure 4

- #1 START
- #2 2, 4 BYTES
- #3 BRANCH
- #4 6, 8 BYTES
- #5 END
- 101 SET INSTRUCTION ADDRESS
- 102 IF1 FOR THREE TIMES
- 103 INSTRUCTION STORAGE POINTER 00, 01
- 104 SUCCESSIVE INSTRUCTION POINTER 000, 001
- 105 STORE INSTRUCTION IN ANALYSIS REGISTER
- 106 INDICATE TO BRANCH TO INSTRUCTION CODE
- 107 ADD SUCCESSIVE LOCATION AND INSTRUCTION WORD LENGTH
- 108 INPUT RESULT OF ADDITION IN SUCCESSIVE LOCATION POINTER

Japanese Publication No. 60-117335  
(translation)

- 109 INPUT PREVIOUS LOCATION IN CURRENT INSTRUCTION LOCATION  
POINTER
- 110 ANALYZE CURRENT INSTRUCTION
- 111 HOW LONG INSTRUCTION WORD?
- 114 IF1 FOR THREE TIMES
- 115 READ SUCCESSIVE OPERAND
- 116 CALCULATE PRECEDENT OPE ADDRESS



⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A) 昭60-117335

⑬ Int.Cl.<sup>4</sup>

G 06 F 9/38  
9/28

識別記号

庁内整理番号

B-7361-5B  
A-8120-5B

⑭ 公開 昭和60年(1985)6月24日

審査請求 未請求 発明の数 1 (全6頁)

⑮ 発明の名称 情報処理装置

⑯ 特 願 昭58-224130

⑰ 出 願 昭58(1983)11月30日

⑱ 発 明 者 勇 恵 一 尾張旭市曙丘町池上1番地 株式会社日立製作所旭工場内  
⑲ 発 明 者 常 広 隆 司 国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中  
央研究所内  
⑳ 発 明 者 中 越 順 二 国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中  
央研究所内  
㉑ 発 明 者 中 井 幸 一 尾張旭市曙丘町池上1番地 株式会社日立製作所旭工場内  
㉒ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地  
㉓ 代 理 人 弁理士 高橋 明夫 外1名

明 願 書

1. 発明の名称 情報処理装置

2. 特許請求の範囲

(1) 先行制御のための複数の命令バッファ・レジスタと、該レジスタへの次命令格納位置を指示するポインタと、該レジスタからの読み出し位置を指示するポインタとを有する情報処理装置において、上記命令バッファ・レジスタから現在命令および後読命令の読み出し位置を指示する各ポインタと、該後読命令読み出し位置を示すポインタの値および上記次命令格納位置を指示するポインタの値を比較して、次の命令を命令バッファ・レジスタに読み出せるか否かを判別する手段と、上記判別手段により、次の命令の読み出しをキャンセルするか否か決定し、かつ次の命令が分岐命令のときには無条件で読み出すことを指示する制御手段とを有することを特徴とする情報処理装置。

3. 発明の詳細な説明

(発明の利用分野)

本発明は、情報処理装置に関し、特に性能を低下させずにハードウェア量の削減が可能なマイクロプログラムによる先行制御型情報処理装置に関するものである。

(発明の背景)

情報処理装置の動作を高速化するために、先行制御が行われているが、その場合、命令語を先読みするための命令語読み出しレジスタのデータ長は、(命令語最大語長) × (先読み回数 + 1) だけ必要である。

すなわち、従来の先行制御型情報処理装置の命令バッファ・レジスタは、例えば、第1図に示すように単位命令レジスタIR1～IR4で構成されている。第1図において、L1は最小命令語長、L2は記憶装置のデータ読み出し長、L3は最大命令語長である。

このように、命令バッファ・レジスタの構成は、最小命令語長L1のn倍(ここでnは正の整数)のデータ長L2(記憶装置のデータ読み出し長)を有するレジスタIRを複数本備え、かつ命令バ

ツファ・レジスタの全データ長は最大命令語長 $L_3$ の $m$ 倍(ここで $m$ は正の整数)の長さを有している(第1図の場合は、 $m = 2$ である)。これによつて、最大命令語長 $L_3$ の命令が連続した場合でも、次の命令が命令バッファ・レジスタ内に格納されているため、先行制御が可能である。

しかし、最大命令語長 $L_3$ の命令の使用頻度が低い情報処理装置においては、命令バッファ・レジスタの全データ長が最大命令語長 $L_3$ の整数倍あつても、それより命令語長が短かく、使用頻度の高い命令に対しては、処理速度を向上させる効果は全くないため、命令バッファ・レジスタが十分に活用されていないという欠点があつた。

(発明の目的)

本発明の目的は、このような従来の欠点を解消するため、ハードウェア量を低減して小型化、低価格化を図るとともに、使用頻度の低い命令語長の長い命令に対しても性能を低下させない情報処理装置を提供することにある。

(発明の概要)

(3)

本実施例では、第2図に示すように、最小命令語長 $L_1$ が2バイト、命令読み出し語長 $L_2$ が4バイト、最大命令語長が8バイト、命令バッファ・レジスタの全データ長が12バイトの命令バッファ・レジスタを用いる。この場合には、第1図の従来の命令バッファ・レジスタに比べてハードウェア量は $1/4$ だけ削減されている。

第3図には、第2図の命令バッファ・レジスタ2が設けられている。第3図において、1は主記憶装置、3は命令セクタ、4は命令解析レジスタ、5は命令語長デコーダ、6、8はいずれも演算器、7は次命令の格納位置ポインタ、9は先頭位置を示す後読命令読み出し位置ポインタ、10は先頭位置を示す現在命令読み出し位置ポインタ、11はマイクロ命令レジスタ、12は主記憶装置制御回路、13は命令読み出し抑止条件回路である。

先ず、命令バッファ・レジスタ2は、命令格納ポインタ7が示す位置に、主記憶装置1から読み出された2命令(現在命令、後読命令)を格納す

(3)

上記目的を達成するため、本発明の情報処理装置は、先行制御のための複合の命令バッファ・レジスタと、該レジスタへの次命令格納位置を指示するポインタと、該レジスタからの読み出し位置を指示するポインタとを有する情報処理装置において、上記命令バッファ・レジスタから現在命令および後読命令の読み出し位置を指示する各ポインタと、該後読命令読み出し位置を示すポインタの値および上記次命令格納位置を指示するポインタの値を比較して、次の命令を命令バッファ・レジスタに読み出せるか否かを判別する手段と、上記判別手段により、次の命令の読み出しをキャンセルするか否かを決定し、かつ次の命令が分岐命令のときには無条件で読み出すことを指示する制御手段とを有することに特徴がある。

(発明の実施例)

以下、本発明の実施例を、図面により説明する。

第2図は、本発明が用いる命令バッファ・レジスタの構成図であり、第3図は本発明の一実施例を示す情報処理装置のブロック図である。

(4)

る。命令格納ポインタ7は、2ビットで構成されるので、“00”、“01”、“10”により命令バッファ・レジスタ2の3つの命令格納位置( $L_2$ 単位)を指定できる(ここでは、“11”は使用されない)。

そして、命令バッファ・レジスタ2に命令が格納された後、演算器6により1を加算して格納位置を更新する。この場合、演算器6は1加算回路であつて、加算結果が“11”となつたときには、“00”を出力するようになっている。

命令セクタ3は、現在命令位置ポインタ10と、後読命令位置ポインタ9によつて指定された命令バッファ・レジスタ2内の命令をそれぞれ選択し、各命令語を順次取り出して次段の命令解析レジスタ4に格納する。命令が命令解析レジスタ4に格納されると、命令語長デコーダ5は、命令解析レジスタ4の命令語長ビットをデコードし、後読命令の命令語長を生成して加算器8に送出する。

後読命令位置ポインタ9は、3ビットで構成さ

れ、マイクロ命令が次の命令語を実行するとき、演算器8により命令語長デコード5で生成した後読命令の命令語長を加算して、その結果を後読命令位置ポインタ9に出力する。これに先立つて、後読命令位置ポインタ9に格納されている内容も、現在命令位置ポインタ10に移すと同時に、加算器8の他の入力として加える。

後読命令位置ポインタは、"000", "001", "010", "011", "100", "101"により、命令バッファ・レジスタ2の8つの命令格納位置(11単位)を指定できる(ここでは、"110", "111"は使用されない)。演算器8は、加算結果が"110"となつたときには、"000"を出力する。この場合、3ビットのうちの上位2ビットが素数で区切られた12単位の位置を示し、下位1ビットが素数で区切られた11単位の前後の位置を示している。

主記憶装置1は、1つのアドレスに4バイト(32ビット)単位で格納されているが、命令の単位としては2バイト(16ビット)、4バイト、

(32ビット)および8バイト(64ビット)の範囲があるので、主記憶装置1から現在命令とともに後読命令を先読みする場合、命令バッファ・レジスタ2に2つの命令がすべて格納されるときと、後読命令の一部のみしか格納されないで、残りの部分が主記憶装置1にまだ格納されているときがある。

本実施例では、(1)すでに先読みした命令がすべて命令バッファ・レジスタ2に格納されていて、その後読命令を実行する場合(通常のステップ命令のとき)と、(2)先読みした命令がすべて格納されていても、その命令をキャンセルして同じ位置に新たに読み出しを行う場合(次の命令が分岐命令のとき)と、(3)先読みされる命令が命令バッファ・レジスタ2にすべて格納できないので、現在命令をマイクロ命令が実行している間に、現在命令の位置に後読命令を読み込んだ後、後読命令のアドレス計算を行ってしまう場合(バッファ・レジスタ量が不足するとき)のいずれにおいても、著しく性能を低下させずに実現することができる。

(7)

(8)

また、第3図においては、マイクロ命令により先行制御が行われており、通常、無条件分岐の場合には、マイクロ命令レジスタ11のアドレス部を制御記憶アドレス・レジスタに移すとともに、マイクロ命令レジスタ11の特定の1ビットを分岐指定ビットとして"0"または"1"をセットする。条件付き分岐の場合には、分岐テストの結果、条件が成立したときマイクロ命令レジスタ11のアドレス部を制御記憶アドレス・レジスタに移し、また条件が不成立ならば制御記憶アドレス・レジスタの内容を+1して正常ステップの制御に移る。記憶装置制御回路12は、マイクロ命令レジスタ11の命令読出し制御部Aの1ビットが"0"のとき、無条件に主記憶装置1より命令読み出しを行うように制御する(分岐命令の場合)。このマイクロ命令の指示を、IF1(命令フェッチ1)と呼ぶことにする。また、マイクロ命令レジスタ11の命令読出し制御部Aの1ビットが"1"のとき、命令読出し抑止条件回路13の出力が"0"の場合には命令読み出しを行い、"1"の場合には命

令読み出しをキャンセルする制御を行う。上記のようなキャンセルを行うマイクロ命令の指示を、IF2(命令フェッチ2)と呼ぶことにする。

命令読出し抑止条件回路13は、命令格納ポインタ7の2ビットと後読命令位置ポインタ9の3ビットのうちの上位2ビットを比較し、一致しているとき"1"を出力し、それ以外の場合"0"を出力する。ここで、命令読出し抑止条件回路13の出力が"1"のときとは、すでに後読命令が格納されている位置に、次の命令読出しの格納位置が一致していることを意味する。つまり、マイクロ命令がIF1を指示したときは、後読命令が格納されている位置に次の命令語を格納することをキャンセルし、新たに格納された命令(例えばブランチ命令等)より実行することを意味する。また、マイクロ命令がIF2を指示したときは、後読命令が格納されている位置に、次の命令語を格納することをキャンセルして、すでに格納されている上記後読命令より実行することを意味している。

第4図は、第3図の動作を制御するマイクロプ

(9)

(10)

プログラムのフローチャートである。

先ず、先頭の命令を読み出すため、マイクロ命令は命令アドレスをアドレス・レジスタ(図示省略)にセットし、IF1を3回指示する(12単位の命令バッファ・レジスタが3個存在するため、ステップ101, 102)。このとき、命令格納ポインタ7はアドレス・レジスタがセットされたとき"00"とし、命令バッファ・レジスタ2の先頭より命令語を順次格納する(ステップ103)。また、後続命令ポインタ9は、命令アドレスが偶数のとき"000"を、奇数のとき"001"をそれぞれセットする(ステップ104)。命令が命令バッファ・レジスタ2に読み出されて格納された後、マイクロ命令は命令コード・ブランチを指示し、命令を実行するマイクロ命令の先頭マイクロアドレスにブランチする(ステップ106)。なお、命令コード・ブランチの前に、命令セクタ3は後続命令位置ポインタ9が示す位置より命令を読み出し、命令解析レジスタ4に格納する(ステップ105)。命令語長デコーダ5は、命令語

#### (11)

ステップ113)。すなわち、すでに格納されている後続命令が4バイト長以下のときは、残りのレジスタに読み出して格納できるのでIF1を発行し、8バイト長以上のときには残りのレジスタに格納できないのでIF2を発行して、次の読み出しをキャンセルできるようにする。

(11)命令語長によらず、分岐命令のときには、IF1を3回指示する(ステップ114)。この場合には、無条件で分岐先の命令を読み出して格納しそれを実行する。3回指示するのは、12単位の命令バッファ・レジスタが3個あるからである。

マイクロ命令が現在の命令を実行する場合に、命令解析レジスタ4の内容を参照しなくなつたとき、つまり次の命令語長がレジスタの残りの部分より大のため、次の命令が読み出せないときには、IF1またはIF2を指示し、後続命令のオペランド・アドレス計算、オペランド読み出しを指示する(ステップ115)。この場合、後続命令はIF2を指示しているため、次の命令読み出しにより破棄されることなく、先行アドレス計算指示

長をデコードし、演算器8に命令語長を出力することにより、演算器8は前にセットされている後続命令位置ポインタ9の内容と命令語長を加算する(ステップ107)。後続命令位置ポインタ9は、マイクロ命令が命令コード・ブランチを指示したとき、この加算結果を格納する(ステップ108)。これと同時に、現在命令位置ポインタ10は、前の後続命令位置ポインタ9の内容を格納する(ステップ109)。命令コード・ブランチした先のマイクロ命令は、その命令語長にしたがつて、IF1、IF2を次のとおりに指示するとともに、現在の命令を実行する指示を行う(ステップ110)。

(1)命令語長が2バイトまたは4バイトのとき、IF2を1回指示する(ステップ112)。すなわち、すでに格納されている後続命令が最大語長13であつても、4バイト以下ならば次の命令を格納できるからである。

(2)命令語長が6バイトまたは8バイトのとき、IF1とIF2をそれぞれ1回ずつ指示する(ス

#### (12)

によつて、命令セクタ3は後続命令位置ポインタ9が示す位置より命令バッファ・レジスタ2内の後続命令を命令解析レジスタ4に格納し、後続命令のオペランド・アドレス計算を行い、計算されたアドレスにしたがつてオペランドを読み出す。ここで、現在命令が2バイト長のときには、命令読み出しの終了を待たずに、後続命令の先行オペランド・アドレス計算が可能であるが、現在命令が4バイト長で、かつ後続命令が6バイト長以上のとき、命令バッファ・レジスタ2には命令の一部しか格納されていないため、命令読み出しの終了を待つて、後続命令の先行オペランド・アドレス計算を行う(ステップ116)。

現在命令が6バイト長、8バイト長の場合も、前と同じようにして、後続命令が命令バッファ・レジスタ2に一部しか格納されていないとき、命令読み出しの終了を待つて、後続命令の先行オペランド・アドレス計算を行う。

上記の後続命令が一部しか格納されていない場合に、命令読み出しの終了を待つ間、マイクロ命

令は現在の命令を実行することにより、実行待ち時間の減少を図る。

このように、本実施例では、命令バッファ・レジスタの全データ長を短くして、コスト低下を図り、命令語長が短い命令に対しては高速動作させる一方、命令語長の長い命令が連続した場合にも、それほど性能を低下させることなく命令を実行することができる。

(発明の効果)

以上説明したように、本発明によれば、命令バッファ・レジスタの全データ長を、最大命令語の $n$ 倍(例えば2倍)より短かく、かつ命令読み出し長の $m$ 倍(例えば3倍)のデータ長にすることができるので、ハードウェア量が削減される結果、小型化、低価格化が可能となり、しかも、従来に比べて性能の低下を避けることができる。

4. 図面の簡単な説明

第1図は従来における命令バッファ・レジスタの構成図、第2図は本発明における命令バッファ・レジスタの構成図、第3図は本発明の一実施例を

示す情報処理装置の要部ブロック図、第4図は第3図の制御を行うマイクロプログラムの動作フローチャートである。

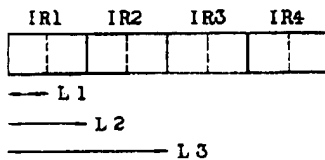
1:主記憶装置、2:命令バッファ・レジスタ、3:命令セクタ、4:命令解析レジスタ、5:命令語長デコーダ、6, 8:演算器、7:命令格納ポインタ、9:現在命令位置ポインタ、10:後続命令位置ポインタ、11:マイクロ命令レジスタ、12:記憶装置制御回路、13:命令読み出し抑止条件回路。

代理人 弁理士 高橋 明 夫

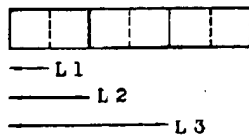
Q5

Q6

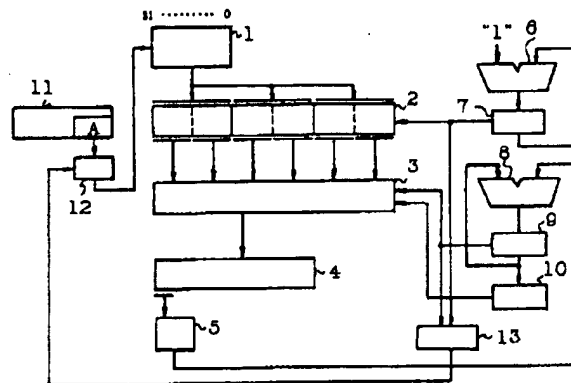
第 1 図



第 2 図



第 3 図



第 4 図

